DOCUMENT 170-04

DATA REDUCTION AND COMPUTER GROUP (DR&CG)

GROWTH AND MIGRATION PLAN FOR DATA INSIGHT

WHITE SANDS MISSILE RANGE
REAGAN TEST SITE
YUMA PROVING GROUND
DUGWAY PROVING GROUND
ABERDEEN TEST CENTER
NATIONAL TRAINING CENTER
ELECTRONIC PROVING GROUND

NAVAL AIR WARFARE CENTER WEAPONS DIVISION
NAVAL AIR WARFARE CENTER AIRCRAFT DIVISION
NAVAL UNDERSEA WARFARE CENTER DIVISION, NEWPORT
PACIFIC MISSILE RANGE FACILITY
NAVAL UNDERSEA WARFARE CENTER DIVISION, KEYPORT

30TH SPACE WING
45TH SPACE WING
AIR FORCE FLIGHT TEST CENTER
AIR ARMAMENT CENTER
AIR WARFARE CENTER
ARNOLD ENGINEERING DEVELOPMENT CENTER
BARRY M. GOLDWATER RANGE
UTAH TEST AND TRAINING RANGE

NATIONAL NUCLEAR SECURITY ADMINISTRATION (NEVADA)

DISTRIBUTION A: APPROVED FOR PUBLIC RELEASE
DISTRIBUTION IS UNLIMITED

This page intentionally left blank.

DOCUMENT 170-04

GROWTH AND MIGRATION PLAN FOR DATA INSIGHT

DECEMBER 2004

Prepared by

DATA REDUCTION AND COMPUTER GROUP
RANGE COMMANDERS COUNCIL

Published by

Secretariat
Range Commanders Council
U.S. Army White Sands Missile Range,
New Mexico 88002-5110

This page intentionally left blank.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

PREFACE

The information contained in this report is intended for use as a guide to any range or organization requiring knowledge of available technologies, and capabilities of software platforms used for post-test telemetry data reduction analysis.

The Data Reduction And Computer Group (DR&CG) would like to acknowledge the production of this document for the RCC by MathWorks Consulting Services:

Author: Mr. Raymond Norris, Senior Consulting Engineer
Consulting Services
The MathWorks, Inc.
3 Apple Hill Drive
Natick, MA 01760

Telephone: (508) 647-7372/7000
Fax: (508) 647-7037
Email: rnorris@mathworks.com, consulting@mathworks.com
Web: http://www.mathworks.com/consulting

Please direct any questions to:

Secretariat, Range Commanders Council
ATTN: TEDT-WS-RCC
1510 Headquarters Avenue
White Sands Missile Range, New Mexico 88002-5110
Telephone: (575) 678-1107, DSN 258-1107
E-mail usarmy.wsmr.atec.list.rcc@mail.mil

Proprietary Notice:

The MathWorks Consulting
consulting@mathworks.com

MATLAB, Simulink, Handle Graphics, Real-Time Workshop and Stateflow, are registered trademarks of The MathWorks, Inc. (MathWorks). TargetBox is a trademark of The MathWorks, Inc. Other product or brand names are trademarks or registered trademarks of their respective holders. This document contains material that is proprietary to RCC and MathWorks.

This page intentionally left blank.

CHAPTER 1

EXECUTIVE SUMMARY

As part of the Data Reduction and Computer Group (DR&CG) Task-32 (DR-32), this document contains the Growth and Migration Plan for the Data Insight code base. This document describes the areas of Data Insight that need to be developed in order to make it, with MATLAB, the software platform of choice for post-test telemetry data reduction analysis. The document also discusses the manner in which to acquire Data Insight and the cost for the consulting engagement to continue developing Data Insight.

The background for DR-32 is described in the DR-32 Final Report at Appendix A. An excerpt from the Final report is shown below:

"The goal of DR-32 was to begin the initial phase of providing a common telemetry data reduction system that can be used DoD-wide. Given the parameters of several members data reduction/analysis systems, leveraging a replacement for DataProbe, which was common among many members, was viewed as a good start for the initial phase. To that end, this initial phase was successful. However, to the overall goal of providing a common telemetry data reduction system, follow-on tasking was required to go further. The follow-on tasking request to RCC was denied for reasons that the tasking was outside the DR&CG charter. When the follow-on tasking was denied, a modified scope was adopted that maximized the initial phase vice the ultimate goal.

In light of maximizing the initial phase, those members who have embraced MATLAB as its foundation for data analysis and, for those members who needed to find a bridge from DataProbe to MATLAB, both Data Insight and the Telemetry Toolbox provide a "first base" solution. Members can purchase maintenance of Data Insight from MathWorks directly via a GSA Reseller. Members can also contract Raytheon Missile Systems to modify the Telemetry Toolbox to fit their data analysis requirements. The adjective, "first base" denotes that both systems are immature. Much work and funding is required to get them to what one would consider a robust, performing system. Several user communities are using both systems.

Per the deliverables of the modified scope (approved by the DR&CG board), an operable beta release of Data Insight with a Growth & Migration Plan for members to follow was delivered. The definition of operable, however, is somewhat of a misnomer. No member should think that they would, necessarily be able to take the Data Insight version delivered here and proceed to use it effectively without help from MathWorks Consulting Division. Further, participating members are continuing to fund updates via maintenance contracts and, as such, the delivered beta system is not something one should invest time or effort into adopting. The recommendation of this report is that if you want to view the best Data Insight has to offer, contact MathWorks for a beta release of the most recent release. The current MathWorks contact, as of 2004 is Raymond Norris, 508.647.7372, rayn@mathworks.com.

Telemetry Toolbox, and a Growth & Migration Plan for it, was not delivered as the modified scope limited the focus to only Data Insight.

The deliverables of this task do not provide a growth and migration plan to a common telemetry data reduction system that can be used DoD-wide, i.e., a plan that discusses how to bridge from DataProbe to Data Insight/Telemetry Toolbox to a common, open source, telemetry data reduction system. To understand why, it is critical to remember that this task was always viewed as the first step to a lengthy process. The lofty goal of providing a beta system that actually worked to bridge DataProbe users to a PC based system using MATLAB was extremely successful given a $100K investment vice the 10 million dollars originally invested in creating DataProbe. Hence, the Growth & Migration Plan delivered with this task is focused at the granular level of the bridge of DataProbe to Data Insight. It was originally planned to use a portion of the funds to provide a Growth and Migration Plan that addressed the global common picture, but as was stated, a decision was made to maximize the initial phase given the decision by the RCC to not continue this line of tasking.”

CHAPTER 2

INTRODUCTION

The Data Reduction and Computer Group (DR&CG) of the Range Commanders Council (RCC) put forth an initiative (DR-32) to determine the feasibility of using the Data Insight code base, written in MATLAB, to help transition DataProbe users to MATLAB. Data Insight is a set of MATLAB M-files used for importing data from a data source via a DataProbe dictionary. Data Insight has many capabilities, including production of basic plotting routines for time-series objects as well as report generation compatible with Microsoft Word and Microsoft Excel documents. This document fulfills the requirements of the modified scope of Task DR-32 to provide a Growth and Migration Plan for Data Insight.

To accomplish the DR-32 initiative, members of the DR&CG entered into a contract with MathWorks Consulting to acquire, develop, test, document, deploy, train, and support Data Insight. This contractual arrangement provides two unique opportunities. First, it allows interested parties to immediately have access to Data Insight, thereby allowing them start migrating from DataProbe. This arrangement is different than most consulting engagements where the user would need to start from scratch with requirements, development, testing, etc. Second, the MathWorks Consulting group will use tested development approaches towards the development efforts of Data Insight. This approach will allow Data Insight to have a common look and feel of other toolboxes as well as being able to take advantage of a tight coupling with MathWorks Development. In other words, this approach will provide input on Data Insight for development of features needed in the MATLAB language.

The recommended approach is that development of Data Insight be carried out in two phases. The first phase will include creating a vehicle for acquiring Data Insight and providing immediate support for the code base as is, making structural changes where needed, making improvements to the documentation, and providing an updated Data Insight demonstration model. The second phase is a longer term approach to the development of Data Insight, including writing a requirements document, writing a design document, possible support for non-DataProbe dictionaries, improving the MATLAB core features as well as add-on toolboxes, and perhaps a graphical user interface (GUI).

The following chapters provide an outline for implementation of the DR-32 effort:

Chapter 3 Intended users of Data Insight

Chapter 4 A brief development plan

Chapter 5 A brief project management plan

Chapter 6 Product and maintenance plan

Chapter 7 Pricing and purchasing of a Data Insight maintenance plan

Chapter 8 Software rights and warrantees of Data Insight

Chapter 9 Long-term direction of Data Insight

This page intentionally left blank.

CHAPTER 3

INTENDED USERS OF DATA INSIGHT

3.1 Categories of Intended Users

There are three types of intended users of Data Insight; the first two types are post-test engineers[1] and the third type is the System Administrator (SA).

3.1.1 <u>Post-test engineers (non power users of MATLAB)</u>. The first type of intended user is the engineer who is familiar with using a test exercise analysis tool. The user:

a. Typically uses an in-house tool such as Boeing's QuickView to import data, perform analysis, plot and annotate, and create reports.

b. May be familiar with MATLAB, but is not a power user.

c. Is looking for a Commercial Off-The-Shelf (COTS) package that offers the option of minimal usage of the MATLAB Command Window.

3.1.2 <u>Post-test engineers (comfortable with MATLAB)</u>. The second type of intended user is the engineer who performs the same functions as the first, but is much more comfortable using MATLAB. The user:

a. Will most likely want to add to and increase the Data Insight functionality.

b. Will have a solid understanding of the back-end functionality of Data Insight.

c. Does not require a "canned" application due to familiarity with MATLAB, and in many cases, will not use a specific graphical user interface (GUI) application.

3.1.3 <u>System Administrator (SA)</u>: The third type of intended user is the Systems Administrator (SA) for Data Insight. Specifically, the SA:

a. Will be responsible for creating a FileDef structure for a specific DataProbe data dictionary.

b. Should be very familiar with the layout of the dictionary and has at least minimal knowledge of MATLAB in order to write scripts.

c. Could also be one of the above post-test engineers, but most likely the second one.

[1] Data Insight does, however, have the capability to work with growing data sources (i.e. the data is being acquired in real-time).

This page intentionally left blank.

CHAPTER 4

DEVELOPMENT PLAN FOR MATURING DATA INSIGHT

4.1 Planned Actions

The following section is not an exhaustive development plan, but it outlines specific technical issues and plans for the development of Data Insight. MathWorks Consulting has been engaged in research since September 2002 in preparation for supporting Data Insight. Based on code investigation and user feedback, the following subparagraphs describe immediate actions (in priority sequence) that need to be completed.

4.1.1 Develop unit tests. Unit tests will be written and executed via a test harness[2], thereby automating the regression testing process. Both the unit tests and the test harness will be written in MATLAB, a practice that is similar to how MathWorks Development executes Quality Engineering. These tests will ensure no regression bugs are introduced and also determine structural and usability issues. The unit tests will be more beneficial to the user site if real data from the test site is used for testing. The data should be non-classified and nonproprietary so that the data can be shipped as part of the test suite.

4.1.2 Re-architect the code base. After an initial evaluation of Data Insight, several key underlining pieces need to be enhanced. Specifically, Data Insight makes use of the MATLAB *assignin* and *evalin* functions, mutates base workspace variables, does not have key functionality of MATLAB classes, and immediately dumps all of the variables into the MATLAB workspace when opening a data source. Before adding additional features, these and other issues will be addressed by looking at the architecture to determine if any pieces need to be redesigned.

4.1.3 Add missing key features. Requirements are still being gathered in order to determine missing features and their order of importance . At this juncture, the following list outlines possible areas that need to be addressed.

a. Auto convert DataProbe dictionary to MAT-files: Rather than reading a data source via a dictionary, use a MAT-file to improve the speed of loading.

b. Remove latency code: Determine and remove bottlenecks in the source code.

c. Improve helper functions: Perform error handling and write code to improve the robustness of dictionary and utility functions.

d. Add Dictionary Tools: Add, delete, and modify variables in a DataProbe dictionary.

e. Add Time Series math: Add math function to a Time Series object.

f. Improve report generation: Improve the robustness of generating Microsoft Word and Microsoft Excel documents: Consider other report formats, such as PDF and XML.

[2] A test harness is an infrastructure for determining which tests to run, executing them, and providing a status report of success and failure for each of the tests.

4.1.4 Rewrite documentation. The current documentation will be updated to reflect new features as well as the re-architecture. The first pass of the update will modify the document to have the look and feel of MathWorks documentation. This will enable integration of the online document into the MATLAB Help Browser.

4.1.5 Create installer. Currently, Data Insight is installed via a ZIP file. After the installation, the MATLAB path is not updated, and so the user must modify the path. The path modification is done typically through a MATLAB script. Immediate plans would include using InstallShield to create a user friendly, polished installer that would also update the MATLAB path.

4.1.6 Create training material. The current training material consists of documentation and the demonstration script. Training material will be developed using Microsoft PowerPoint for educating the user on creation of a FileDef structure, opening data sources and dictionaries, manipulating raw data vectors, performing time series analysis and associated plotting, generating reports, and other tasks.

4.1.7 Write Requirements and Design Documents. MathWorks Consulting needs to write a formal requirements document for importing telemetry data into MATLAB via a DataProbe dictionary. Several sites have already sent requirement documents, including:

a. Tyndall Air Force Base.

b. Redstone Arsenal Army.

c. Cessna Aircraft.

d. Patuxent River, Naval Air.

e. China Lake, Naval Air.

Collating the requirements and writing the document is imperative to ensure all features and constraints are considered prior to full development of Data Insight.

Once the features have been identified and added to the Data Insight Requirements document, the designs for specific components need to be written. The designs will be reviewed by the Data Insight Review Team (RT) and added to the Data Insight Design (“Design”) document (see Figure 4-1).

4.2 Prioritizing Features and Bug Fixes

All feature requests and bug reports will be sent to MathWorks Consulting (datainsight@mathworks.com) and tracked using Microsoft Excel. The following section describes the process for handling development requests.

4.2.1 Features. After reviewing the request, MathWorks Consulting will advise the requestor whether the feature will be added to the Requirements Document or whether it will be deferred. If the request is accepted, MathWorks Consulting will inform all members of the RT that a request has come in and added to the Requirements Document. The requestor will then work with MathWorks Consulting to write a specification for the feature. If any members wish to add sub-requirements to the proposal, they may do so. Once written, the specification will be submitted to all other RT members for review. Once approved by the RT, the specification shall be added to the Design Document. The approval process is outlined at Figure 4-1.

If MathWorks Consulting determines that the feature will not be added to Data Insight and the requestor wishes, the consulting team can let the RT decide if the feature shall be added or not. If the RT decides in favor of the requestor, then the request will be added to the Requirements Document and the process shall continue as normal. If, however, the RT declines the immediate need for the feature, the feature will be postponed indefinitely.

Once processed, all features will be assigned a priority as decided by the RT.

4.2.2 Bugs. Similar to features, when a bug is submitted, the request shall be sent directly to MathWorks Consulting. Once MathWorks Consulting has reviewed the request, it will contact the requestor, stating whether or not the bug will be fixed in the near future. If MathWorks Consulting determines that the bug will not be fixed, the requestor can submit the request to the RT.

Once processed, all bugs will be assigned a priority as decided by the RT.

4.2.3 Priorities. All feature requests and bugs will be assigned one of four priorities.

A - Signifies that the feature or bug must be added or fixed prior to the release of the next major or minor (but not necessarily beta) release.

B - Signifies that the feature or bug has major priority for the next major or minor release.

C - Signifies that the feature or bug has minor priority for the next major or minor release.

D - Signifies that the feature or bug will not be addressed in the foreseeable future.

As features are added and bugs become fixed, periodically, a conference call will be held to update the RT the status of the bugs and enhancements. The Microsoft Excel spreadsheet will be sent out prior to the conference call to be reviewed. Once work becomes finished, the bugs and enhancements will become reprioritized. In addition, as a release date becomes closer, bugs and enhancements will be triaged in order to maintain a reasonable development cycle.

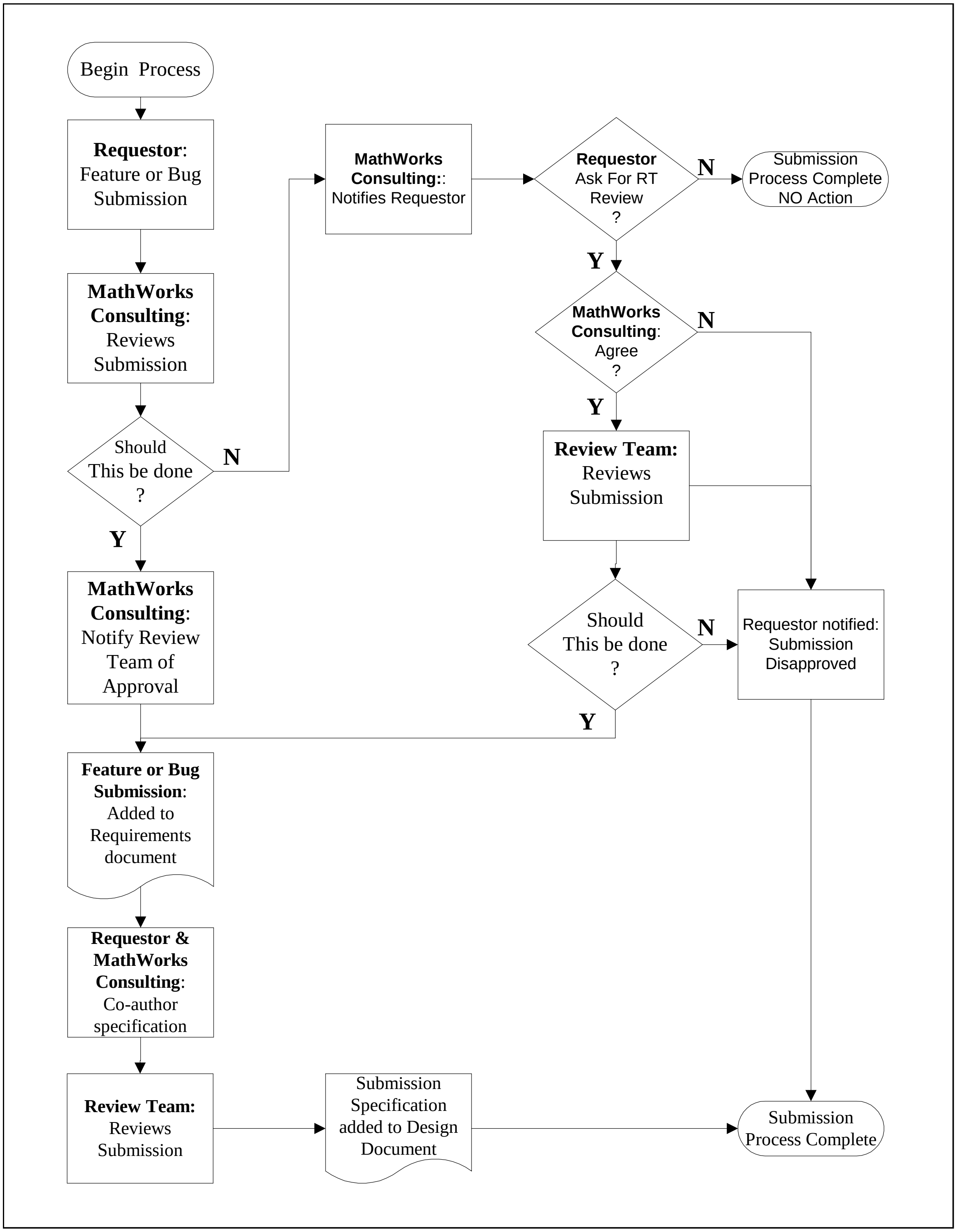
Begin Process
Requestor: Feature or Bug Submission
MathWorks Consulting: Reviews Submission
Should This be done ?
N
Y
MathWorks Consulting: Notify Review Team of Approval
Feature or Bug Submission: Added to Requirements document
Requestor & MathWorks Consulting: Co-author specification
Review Team: Reviews Submission
Submission Specification added to Design Document
MathWorks Consulting:: Notifies Requestor
Requestor Ask For RT Review ?
N
Submission Process Complete NO Action
Y
MathWorks Consulting: Agree ?
N
Y
Review Team: Reviews Submission
Should This be done ?
N
Y
Requestor notified: Submission Disapproved
Submission Process Complete

Figure 4-1. Bug and enhancement submission process.

4.3 Data Insight Review Team

The Data Insight Review Team (RT) is responsible for reviewing requirements and top-level design (specifications) for Data Insight. The team is comprised of:

a. Team Leader (Project Leader of Data Insight, MathWorks Consulting).

b. Data Analysis Tool Development Lead (NUWC, Keyport.)

c. Technical point of contact (TPOC) at each site.

Prior to joining the RT, the TPOC must provide a requirements document outlining features and constraints that the site is looking for in a test exercise analysis tool. The site's requirement document will be reviewed by the Team Leader and brought to the RT if needed. Once reviewed, the requirements will be collated with the Requirements Document. Both the Requirements and Design Documents will be placed under source control, but not shipped with the Data Insight documentation.

The RT is responsible for reviewing new features requests, helping to prioritize features and bug fixes, and reviewing designs for specific features. This will make certain that a feature or bug feature is useful for the entire community not just one site. Conference calls will be held from time to time to discuss the status of the project with the team. The RT will also advise MathWorks Consulting timeframes for releases of Data Insight.

A "majordomo" software program to automate the management of Internet mailing lists will be created for the team.

4.4 Migrating from DataProbe to MATLAB

One of the key capabilities of Data Insight is the ability to use a MATLAB data structure to mimic the open_dat.prb file. The structure was intended to ease migration from DataProbe to MATLAB by assigning the structure's field names to that of characteristics of the DataProbe Dictionary. For example characteristics would include the length of the file header, the units of the record length, and the frame id, etc.

4.5 Connectivity to Third Party Applications

If necessary, capabilities to interface to other applications may be added. These capabilities include, but are not limited to:

a. Data transfer - Once the data has been imported to MATLAB, functionality may be added to export the data to MATLAB-based data files, proprietary or non-proprietary data formats. Examples of data formats include:

1) MATLAB MAT-files,

2) Microsoft Access,

3) Microsoft Excel, and

4) XML

b. Data streaming - MATLAB is capable of calling third party applications via several methods, including, MEX-files, Java, and COM. Therefore, if a third party application has a C/C++/Fortran/Java interface, data could be streamed from MATLAB to it.

Note: Data Insight is an application that runs on MATLAB and requires MATLAB to run.

4.6 MathWorks Consulting's Need to Obtain Copy of NTProbe[3]

A key objective of Data Insight is to help transition DataProbe users to MATLAB. Therefore, it is imperative that MathWorks Consulting be provided a legal copy of NTProbe as well as all relevant documentation in order to:

a. Determine missing features in Data Insight (and MATLAB).

b. Create benchmark test vectors to ensure that Data Insight performs as well, if not better, than DataProbe.

c. To help determine how DataProbe performs Time Series math.

[3] NTProbe is the PC equivalent of DataProbe.

CHAPTER 5

PROJECT MANAGEMENT

5.1 Configuration Control

All source code and support files, such as unit tests, training material, and documentation, will be maintained under revision source control. The revision tool used will be CVS, (http://www.cvshome.org). Although under source control, Data Insight will not be shipped with the CVS repository.

5.2 Release Schedule

First and foremost, the release schedule is dependent on (a) funding and (b) user requirements. Each development cycle will produce a major update and two minor update/bug fix releases. The current tentative schedule as follows:

September 30, 2003:	Ship Data Insight 0.6 beta (minor release)
January 26, 2004:	Ship Data Insight 1.0 (major release)
April 26, 2004:	Ship Data Insight 1.1 (minor release)

The remainder of FY04 will be determined after the release of Data Insight 1.0.

This page intentionally left blank.

CHAPTER 6

PRODUCT AND MAINTENANCE PLAN

6.1 Product and Maintenance Provided by MathWorks Consulting Group

This chapter discusses specifically what is and what is not covered under the Product and Maintenance Plan. The service offering consists of the Data Insight software and a yearly maintenance plan for Data Insight. The software is free of charge, does not run under a license manager, and does not expire. However, the yearly maintenance plan does have a site charge (see Chapter 7) that begins at the start of the fiscal year. The fiscal year starts October 1st of the current year and ends September 30th of the following year. A site is defined to be a group of end users not to exceed 30 users. The group of users must select one person to be the technical point of contact (TPOC). All technical and business questions shall go to and come from the TPOC.

The maintenance shall include the following:

a. Product updates and enhancements - Non-expiring M/P/MEX-file for all source code. The source code may come in the form of M-files, P-files, MEX-files, or other forms such as C and Java. MEX-files will ship for all platforms that Data Insight was developed for (i.e. the current shipping version of MATLAB).

b. MATLAB based test suite - The test suite will contain unit tests for each of the functions (unless deemed unnecessary by MathWorks Consulting) as well as system tests based on beta user-supplied data sources and dictionaries. The tests and test harness will be written in MATLAB.

c. Documentation updates and enhancements - Documentation will be provided for both the source code as well as the test suite. Documentation will be provided in two formats (PDF and HTML). The HTML shall be fully integrated into the MATLAB Help Browser. For the source code, a programmer’s manual and a reference guide will be provided.

d. Known bugs and issues - Each release of Data Insight will include a list of known bugs and issues as well as bugs from previous releases that were fixed. The RT will decide the granularity of which bugs are listed; however, it is likely that from a priority range of **A**, **B**, **C,** and **D** (**A** being the highest priority to fix), that only bugs assigned priorities **A** and **B** would be listed. It is noted that the members of the RT would have a complete list of all bugs and issues.

e. Demos and examples - Demo files, written in MATLAB, will be shipped outlining the key features of Data Insight. In addition, any examples provided in the documentation will be shipped.

f. Access to technical support for Data Insight - The TPOC will be provided support for Data Insight. All Data Insight requests should go to datainsight@mathworks.com. All other MathWorks licensing and product-related issues should continue being sent to support@mathworks.com

g. Training - Any presentations, examples, and exercises used for training will be shipped with Data Insight. Services also include one day of onsite training based on the training material. Though the site is expected to have a working knowledge of MATLAB[4], a quick overview may be shown. The training will last for the duration of one day. The price of the maintenance shall not be changed regardless of whether or not the site uses onsite training. All training material (slides, examples, exercises, etc) will be maintained with each major release and distributed with Data Insight.

All deliverables listed above will be bundled into an InstallShield or similar installer.

6.2 Additional Available Consulting Services

The following sub-paragraphs describe services that are available as an additional consulting engagement and are not part of the core consulting engagement for the development of Data Insight.

6.2.1 Creation of FileDef structure. In order for Data Insight to use a DataProbe dictionary, the site must create a FileDef structure that describes the dictionary. The Data Insight documentation describes in detail how to create this MATLAB structure. A service may be provided to create a specific FileDef structure for the site.

6.2.2 Conversion of DataProbe scripts. Many sites have existing DataProbe scripts. There is no plan to support automatic or manual conversion of these scripts to MATLAB scripts. A service may be provided to translate DataProbe scripts to MATLAB.

6.2.3 Onsite support. Maintenance support does not include onsite support (with the exception of a one-day training which is included in the maintenance plan). A service may be provided for onsite support.

[4] MathWorks Training provides onsite training on MathWorks products including MATLAB and Signal Processing. See www.mathworks.com/training for more information.

CHAPTER 7

PRICING & PURCHASING

7.1 Plan

There is no cost for the installation and use of Data Insight software, including all source code. Data Insight will not expire its software usage and will continue to work with users in good standing depending on changes made to future versions of MATLAB.

Each site is required to subscribe to at least one year of maintenance. The one-year maintenance plan will provide the site with technical support for Data Insight and provide all software updates throughout the U.S. Government fiscal year (October 1 through September 30). Support and usage of Data Insight does not require the site to be on a current subscription plan to MATLAB; however, features and bug fixes will be developed on the latest public version of MATLAB. This does not imply that newer versions of Data Insight will not work on older versions of MATLAB, but that there is no guarantee of backward compatibility. Although no testing will be done on prior versions of MATLAB, if a version of Data Insight is known not to work on a specific version of MATLAB, this fact will be stated in the documentation.

The Review Team (RT) shall be made aware of all known bugs in Data Insight and MathWorks Consulting shall make every reasonable attempt to fix those bugs identified by the largest general population.

7.2 Cost

The cost for one year of software maintenance is $50K per site.

7.3 Purchasing Data Insight Maintenance Plan

A Data Insight maintenance plan can be purchased either by going directly to MathWorks or by using a GSA Schedule. If the capabilities exist, the plan/subscription can be purchased directly through a site's MathWorks Sales representative[5]. A GSA Schedule has been set up with Lyme Computer Systems (http://www.lyme.com) to purchase software maintenance for Data Insight.

Once the purchase order has been received (either directly or via Lyme Computer Systems) all deliverables will shipped either via email or CD to the TPOC. The TPOC should send the contact information shown at Table 7-1 to datainsight@mathworks.com. In addition, the TPOC should send a list of requirements for a data analysis tool.

[5] Data Insight is currently only available in the United States.

TABLE 7-1. TECHNICAL POINT OF CONTACT (TPOC) INFORMATION

Name:	
Title:	
Site:	
Address:	
City:	
State:	
Zip Code:	
E-mail address:	
Business phone number:	
Business fax number:	
Approximate number of intended Data Insight users:	

CHAPTER 8

SOFTWARE RIGHTS AND WARRANTEES

MathWorks owns all intellectual property and software rights to Data Insight. At no time shall these rights be passed on during any deliverable of Data Insight. The warrantees for Data Insight are to be determined (TBD).

This page intentionally left blank.

CHAPTER 9

LONG TERM DIRECTION

The long-term direction of Data Insight clearly points to two commitments. The first commitment is that the MathWorks will continue working with the telemetry community. This commitment is made clear in part by its investment into the continued development of Data Insight. MathWorks Consulting will be using standard development practices of MathWorks, providing a common look and feel of other products developed by MathWorks.

The second commitment is to provide the user with non-expiring source code that currently allows them to use DataProbe dictionaries. Should MathWorks Consulting decide to stop supporting Data Insight, the user will be able to continue with developmental efforts, without the need to rely on MathWorks or MathWorks Consulting.

The two above commitments (MathWorks supported development efforts and immediate availability to non-licensed source code) provide the user with a strong development team for the improvement of Data Insight. These commitments also will dramatically minimize the risk while simultaneously minimizing future risk of any stoppage.

This page intentionally left blank.

APPENDIX A

DR-32 FINAL REPORT DR&CG OF RCC, JUNE 22, 2004

This is the final report for DR-32 Task.

The goal of DR-32 was to begin the initial phase of providing a common telemetry data reduction system that can be used DoD-wide. Given the parameters of several members data reduction/analysis systems, leveraging a replacement for DataProbe, which was common among many members, was viewed as a good start for the initial phase. To that end, this initial phase was successful. However, to the overall goal of providing a common telemetry data reduction system, follow-on tasking was required to go further. The follow-on tasking request to RCC was denied for reasons that the tasking was outside the DR&CG charter. When the follow-on tasking was denied, a modified scope was adopted that maximized the initial phase vice the ultimate goal.

In light of maximizing the initial phase, those members who have embraced MATLAB as its foundation for data analysis and, for those members who needed to find a bridge from DataProbe to MATLAB, both Data Insight and the Telemetry Toolbox provide a “first base” solution. Members can purchase maintenance of Data Insight from MathWorks directly via a GSA Reseller. Members can also contract Raytheon Missile Systems to modify the Telemetry Toolbox to fit their data analysis requirements. The adjective, “first base” denotes that both systems are immature. Much work and funding is required to get them to what one would consider a robust, performing system. Several user communities are using both systems.

Per the deliverables of the modified scope (approved by the DR&CG board), an operable beta release of Data Insight with a Growth & Migration Plan for members to follow was delivered. The definition of *operable*, however, is somewhat of a misnomer. No member should think that they would, necessarily be able to take the Data Insight version delivered here and proceed to use it effectively without help from MathWorks Consulting Division. Further, participating members are continuing to fund updates via maintenance contracts and, as such, the delivered beta system is not something one should invest time or effort into adopting. The recommendation of this report is that if you want to view the best Data Insight has to offer, contact MathWorks and get a beta release of the latest and greatest release. The current MathWorks contact, as of 2004 is Raymond Norris, 508.647.7372, rayn@mathworks.com.

Telemetry Toolbox, and a Growth & Migration Plan for it, was not delivered as the modified scope limited the focus to only Data Insight.

The deliverables of this task do not provide a growth and migration plan to a common telemetry data reduction system that can be used DoD-wide, i.e, a plan that discusses how to bridge from DataProbe to Data Insight/Telemetry Toolbox to a common, open source, telemetry data reduction system. To understand why, it is critical to remember that this task was always viewed as the first step to a lengthy process. The lofty goal of providing a beta system that actually worked to bridge DataProbe users to a PC based system using MATLAB was extremely successful given a $100K investment vice the 10 million dollars originally invested in creating DataProbe. Hence, the Growth & Migration Plan delivered with this task is focused at the granular level of the bridge of DataProbe to Data Insight. It was originally planned to use a portion of the funds to provide a Growth and Migration Plan that addressed the global common picture, but as was stated, a decision was made to maximize the initial phase given the decision by the RCC to not continue this line of tasking.

The rest of this report details the history of the task and consists of the following sections:

- Original Scope Statement
- Modified Scope
- Funding
- Deliverables
- Cost Savings
- Conclusions
- Appended: List of Files associated with the deliverable.

ORIGINAL SCOPE STATEMENT

The following scope statement was from the original statement:

4.1. SCOPE AND SPECIFIC OBJECTIVES: This task will provide a common telemetry data reduction system that can be used DoD-wide. This reduction system will be built with an eye towards a follow-on task of making it generic across any tools platform. In Phase I, as covered by this task, the reduction system will be based on MATLab to leverage off two existing data reduction systems: Data Insight and Telemetry Toolbox. A follow-on task, if so desired, will expand the toolboxes into a generic system. **This task will produce the following deliverables:**

- **Growth & Migration Plan:** This document will determine the capability of Data Insight and Telemetry Toolbox in meeting the data requirements of most DR&CG members for telemetry analysis. Discussions will include:
    - How can these toolboxes be matured to provide a stable product line for RCC Members?
    - How can the toolboxes allow multiple analysis platforms to provide data to and receive data from the toolboxes.
    - Migration Roadmap for RCC Members
    - Software Rights & Configuration Control
    - Software Distribution
    - Funding, Costs, Schedule

- **Telemetry Toolbox & Data Insight - Usable Demonstration Systems:** RMS and TREX will provide unclassified demonstration packages that provide the following functionality:
    - Data Dictionary (add, delete, edit)
    - Data Dictionary Conversion (capability of converting or reading Probe Dictionaries)
    - Data Induction Process/System (akin to Probe Flexible File Server or Probe “FIX” Templates).
    - Documentation. RMS & TREX will provide documentation that details the use and interface to the toolboxes such that member ranges can interface and use their own data sources with the toolboxes.

Demonstration: RMS and TREX will provide demos of the systems at either the November 2002 or May 2003 DR&CG Meeting as appropriate.

MODIFIED SCOPE

Due to the lag time between when the task was proposed and when funds were actually given, a key event changed the complexion of the task. When the contract for DR-32 was being written, we received word that Pt. Mugu, Tyndale AFB, and Edwards AFB were more interested in Data Insight than Telemetry Toolbox. Further, at the DR&CG meetings, the interest level was dramatically in favor of

Data Insight. **After conferring with the DR&CG board**, it was decided to only fund the MathWorks portion of the deliverable (~$67K).

Second Change to Tasking: It was recommended and approved by the DR&CG board that we use the remaining funds to address some of the significant beta release issues to make Data Insight more robust. Based on that approval, Jim Tedeschi, the task funds manager, approved and extended the funding date. This proposal and decision was in line with the original scope of the task, i.e., a working beta release product. NUWC Keyport, NUWC Newport, and Redstone Test Arsenal are currently using Data Insight in production. Tyndal AFB is in negotiations. Programmatic issues diminished Edwards AFB's requirements for DataProbe capability. Other sites, Eglin AFB, Pt Mugu, China Lake, and Patrick AFB among them, originally expressed interest in Data Insight and may or may not use it.

FUNDING

$100K of CTEIP ARTM funds was received in April '03 for DR-32. Per the scope above, MathWorks (they bought Data Insight from TREX) and RMS were to receive funds to create a Growth and Migration Plan and a viable beta release of both systems.

DELIVERABLES

- The Growth & Migration Plan was delivered on schedule at the August 2003 DR&CG meeting at NUWC Keyport.
- Two releases of Data Insight were delivered, one in September 2003 and one in December 2003. In September, an Issues and Performance Report was delivered by MathWorks with respect to the original beta release of Data Insight.
- This report, The Growth Migration Plan, a zip file of the Beta Data Insight, the September and December Issues & Performance Report have been submitted to the DR&CG Secretary, Susan Wood.

COST SAVINGS

Each participating member may report cost savings equal to the cumulative funding of Data Insight. To date, each member except NUWC Keyport may claim $225K in savings. NUWC Keyport has claimed $100K in savings.

CONCLUSION

An operable beta release of Data Insight with a Growth & Migration Plan for members to follow was delivered per DR-32 requirements. Three members, NUWC, Redstone Arsenal, and Tyndall AFB are currently active. Others may follow...for some members the need for DataProbe replacement has been overcome by events.

DR-32 DELIVERABLES:

DI_10_06_02 – Source Code.zip
Working Beta System plus Source Code

DI_r0p6_2003_09_30.zip
Bonus: Updated Beta System sans Source Code

RCC_DataInsight_G&M_Plan_2003_08_27.ppt
Purpose: Growth & Migration Plan presented to DR&CG. Tells what was accomplished, what needs to be accomplished, and how to acquire maintenance/updates.

Data Insight Beta Issues and Performance Sep-2003
Data Insight Issues and Performance MarkedUp wrt Sep-Nov 2003.xls
Data Insight Issues and Performance November 2003.xls
Purpose: Status on Data Insight Development

DataInsight0.3.doc
Purpose: Original User's Manual, as purchased from TREX

DataInsightGrowthandMigrationPlan.doc
Purpose: Growth and Migration Plan

SAMPLE DATA SETS & DICTIONARIES

SampleFileDef.m; .p
Purpose: Sample Data File Descriptor file. Use to denote characteristics of input raw data files.

Test14.cod; .dic; .fdt
Purpose: Sample data dictionary. The .cod contains coded variables. The .dic contains the general description of variables. The .fdt contains specific unpacking directions for each variable.

Test14.didic
Purpose: A Data Insight compilation of the .cod, .dic, and .fdt information

Test14.png; .tel; Test14xx.TEL
Purpose: Sample data sets

Test14.tel.dimapfile
Purpose: A Data Insight mapped file of the sample data set.

Test14filedef2.m; .p; Test14filedef.m; .p; Test14filedef_testtime.m; .p
Purpose: File descriptors for the sample data set.

SUPPORT DOCS:

DR32 Final Report 06.22.04

RCC_DataInsight_StatusReport_2003_08_27.ppt
Purpose: Report Status of Data Insight as of August 27, 2003
Highlights: GSA Schedule via Lyme Computer

RCC_Telemetry Task Proposal Growth_2003_08_27.ppt
Purpose: NON-FUNDED task proposal

DR-32 DELIVERABLES:

DI_10_06_02 – Source Code.zip

Working Beta System plus Source Code

DI_r0p6_2003_09_30.zip

Bonus: Updated Beta System sans Source Code

RCC_DataInsight_G&M_Plan_2003_08_27.ppt

Purpose: Growth & Migration Plan presented to DR&CG. Tells what was accomplished, what needs to be accomplished, and how to acquire maintenance/updates.

Data Insight Beta Issues and Performance Sep-2003
Data Insight Issues and Performance MarkedUp wrt Sep-Nov 2003.xls
Data Insight Issues and Performance November 2003.xls

Purpose: Status on Data Insight Development

DataInsight0.3.doc

Purpose: Original User's Manual, as purchased from TREX

DataInsightGrowthandMigrationPlan.doc

Purpose: Growth and Migration Plan

SAMPLE DATA SETS & DICTIONARIES

SampleFileDef.m; .p

Purpose: Sample Data File Descriptor file. Use to denote characteristics of input raw data files.

Test14.cod; .dic; .fdt

Purpose: Sample data dictionary. The .cod contains coded variables. The .dic contains the general description of variables. The .fdt contains specific unpacking directions for each variable.

Test14.didic

Purpose: A Data Insight compilation of the .cod, .dic, and .fdt information

Test14.png; .tel; Test14xx.TEL

Purpose: Sample data sets

Test14.tel.dimapfile

Purpose: A Data Insight mapped file of the sample data set.

Test14filedef2.m; .p; Test14filedef.m; .p; Test14filedef_testtime.m; .p

Purpose: File descriptors for the sample data set.

SUPPORT DOCS:

DR32 Final Report 06.22.04

RCC_DataInsight_StatusReport_2003_08_27.ppt
Purpose: Report Status of Data Insight as of August 27, 2003
Highlights: GSA Schedule via Lyme Computer

RCC_Telemetry Task Proposal Growth_2003_08_27.ppt
Purpose: NON-FUNDED task proposal

This page intentionally left blank.

APPENDIX B

MIGRATION FROM VAX/PROBE AND NTPROBE TO DATA INSIGHT

1.1 Creating a New File Definition

The following is an excerpt from the Chapter 4 of the Data Insight User’s Guide.

1.1.1 New file definition. Data Insight is designed to handle arbitrary data file formats. When connecting to a new data source, a file definition structure (FileDef) that describes the file format must be created. This section describes how to create a new FileDef and connect to a new data source. Once a FileDef is created it can be used for any data source that conforms to the file format described by the FileDef.

1.1.2 Data Insight features. Some of the key features of Data Insight are:

a. Ability to connect to a data file that contains many types of recorded data,
b. Ability to connect directly to your data files without having to reformat or restructure the data, and
c. Ability to access the data without loading the complete data file into the MATLAB workspace.

1.1.3 File structure overview. Data Insight can access data sources that conform to the generalized notion of recorded frames and records. The FileDef structure identifies the particular format that the data source uses. Figure B-1 shows an outline of the file structure and Table B-1 shows the field definitions.

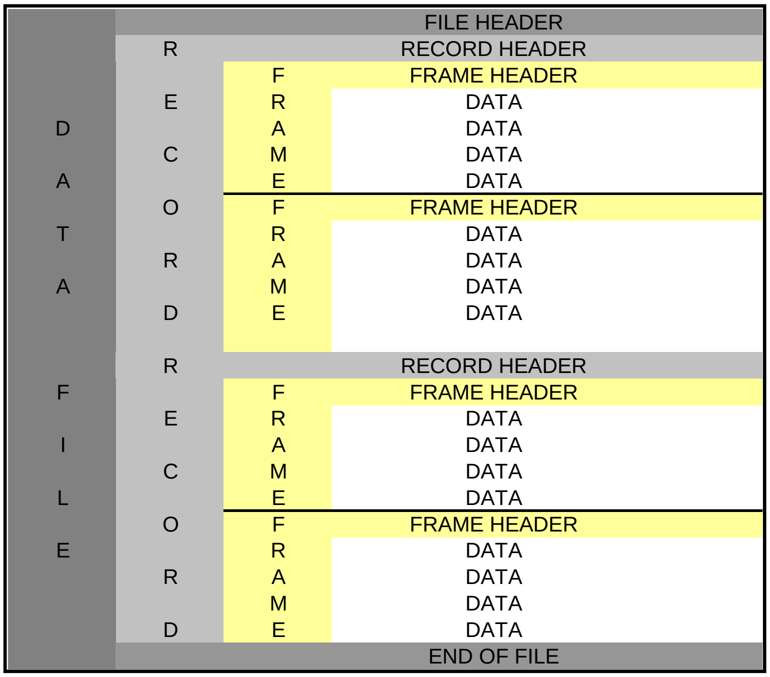

Figure B-1. Basic file structure.

TABLE B-1. DATA FIELDS FORMAT (DATA INSIGHT)

Field Name	**Description**
File Header	A header that is located at the beginning of the file. The file header may contain information about the recorded data, such as the source and the event ID (a unique identifier describing the run number or experiment number of the recorded data).
Record	A block of recorded data. The concept of Record originates from tape I/O, where one record is the buffer of data returned by one I/O routine. This notion of Record can be extended to other media types, where the Record size is set to an efficient amount of data for one I/O routine.
Record Header	A header that is included at the start of the Record that describes the Record. Information commonly found in the header includes the Record length.
Frame	A grouping of data that is recorded at the same time.
Frame Header	A header that is included at the start of each frame that describes the frame. Information commonly found in the frame header includes the frame length, the time that the frame was recorded, and a frame ID that identifies what type of data is recorded in the frame.
Additional information that may be found in the file includes the following:	
Source	The source of the recorded data. For complex systems, there may be more than one source. For example a missile system may record from a Radar source and a Guidance source for each missile test.
EventID	A unique identifier describing the run number or experiment number of the recorded data.
Record Length	The length of recorded Records. This length may be fixed, or may be contained in the Record header or file header.
Frame Length	The length of recorded frames. This length may be fixed, be the same as the Record length, or be contained in the frame header or file header.
Frame ID	An ID describes what type of data is recorded in the frame. Since data in a frame is recorded at the same time, the data in the frame is likely to fall into logical groupings. For example dynamics data from a weapon autopilot (roll, pitch, yaw, roll rate, pitch rate and yaw rate) may be recorded together at a high sample rate in one frame type, while navigational data (X, Y, Z) may be recorded at a lower sample rate in a second frame type. The frame IDs for these two frames could be 1 and 2 respectively.

1.1.4 Creating a FileDef. It is usually convenient to create a function that returns a FileDef. An example of a FileDef script (samplefiledef) can be found in the *DataInsight\Data* directory. The details of the FileDef are described in the next section.

2.1 FileDef Detailed Description

2.1.1 Function Definition and Include File. The first section of the FileDef function simply defines the function and calls includefileopendef. The includefileopendef module defines constants to simplify the creation of FileDef. The following is an excerpt of the samplefiledef:

```
function FileDef = samplefiledef

% Include constants used for assigning FileDef
includefileopendef
```

2.1.2 General Information about the Data File. The first section of the FileDef structure specifies general information about the data file, including the type of file (static or dynamic) and the machine format that the file is recorded in. The following subparagraphs explain.

ϖ*FileDef.Static_File*: The static file field specifies whether the file is static (i.e. the file is no longer being recorded to during analysis) or growing (i.e. the file is still being recorded to while be analyzed real-time).

Values:	
true, false	This field is true if the file is static and false if the file is growing. For growing files, Data Insight will always check the file for data whenever a data object is referenced. For static files, Data Insight will stop checking once EOF is encountered.
Default	true

ϖ*FileDef.Machine_Format*: The Machine_Format field specifies the format of the data file. This field allows Data Insight to be run on a different machine than the machine recording the data. For example, the recorded data may be generated by an embedded processor that uses big-endian byte ordering and then be analyzed using Data Insight on a little-endian machine.

Values:	
'ieee-le'	IEEE floating point with little-endian byte ordering
'ieee-be'	IEEE floating point with big-endian byte ordering
'vaxd'	vax d floating point with VAX byte ordering

'VAXG' ....VAX G FLOATING POINT WITH VAX BYTE ORDERING

'cray'	Cray floating point with big-endian byte ordering
Default	Native machine format

EndOfData: Specifies the position in the file of the end of data (in bytes). This field, in conjunction with FileDef.File_Header.Length (defined below) can be used to access multiple data recordings stored in one physical file.

Default -1 (end of the file)

Example: The sampleFileDef is set up for a static file that is recorded in VAX D floating point and VAX ordering:

```
% FileDef information
FileDef.staticFile = true;
FileDef.Machine_Format = 'vaxd';
```

2.1.3 Source and EventID. Each recorded data file needs to have a source (the source specifies the origin of the file) and an event ID (the run or experiment number of the file). These fields allow for simultaneously opening multiple data files and referencing variables by the source and event ID. Both the source and event ID can be hard-coded (i.e. user specified at file open) or can be read from the file header. If the source or event ID is read from the file header, the standard extract information must be specified. A description of the extract information can be found in the Standard Extract Information paragraph.

ϖ*FileDef.Source.Specified*: The Source.Specified field indicates whether the source of the data file is specified by the user at file open, or specified in the header of the data file.

Values:

From_File_HeaderDataInsight	will read the source from the file header.
User_Specified	User specifies source when calling openDIData
Default	User_Specified

Note: opendatasource will use 'DAT' by default if FileDef.Source.Specified is set to User_Specified, yet not supplied.

2.1.3.1 FileDef.Source Extract Information. If .Source is assigned From_File_Header, standard extract fields are defined as described in Standard Extract Information.

ϖ*FileDef.EventID.Specified*: String or number indicating the file's EventID.

Values:

From_File_Header	DataInsight will read the EventID from the file header.
User_Specified	User specifies EventID when calling openDIData
Default	User_Specified

Note: Opendatasource will use **1** by default if FileDef.EventID. Specified is set to User_Specified, yet not supplied.

2.1.3.2 *FileDef.EventID* Extract Information: If .EventID is assigned From_File_Header, standard extract fields are defined as described in Standard Extract Information paragraph.

Example:
In `samplefiledef`, the source is read from the file header, while the event ID is user specified. The user will specify the event ID when opening the data source with opendatasource.

```
% Source and event ID
FileDef.Source.Specified = From_File_Header;
FileDef.Source.Start_Word = 2;
FileDef.Source.Start_Bit = 0;
FileDef.Source.Field_Length = 32;
FileDef.Source.Data_Type = 'uchar';

FileDef.EventID.Specified = User_Specified;
```

3.1 File Header.

If the data file contains a header, it is necessary to specify the length of the header. This length can be a fixed length or a length read from the header itself. A length of zero can be used to specify that the file does not have a header. If the file header length is read from the file header, the standard extract information must be specified. A description of the extract information can be found Standard Extract Information. The extraction information for the file header length allows for a special case in which the file header length is specified in units of records. In this case, the length is determined using the information in the `FileDef.Record_Header` fields.

ϖ *FileDef.File_Header.Length*: Number that indicates the length of the file header. The default units are Bytes. If the units used in the recorded file are not Bytes, use the FileDef.File_Header.Scale field to convert the to convert recorded units to units of Bytes.

Values:

From_File_Header:	DataInsight will read the length from the file header
No_File_Header:	Sets the file header to a length of zero.
[integer >=0]	Sets the file header to the length specified. Units are in Bytes, unless the FileDef.File_Header.Scale is set to indicate the units.
Default (0)	By default, no file header is assumed.

ϖ *FileDef*.File_Header extract info: If FileDef,File_Header.Length is set to From_File_Header(1), standard extract fields are defined as described in the Standard Extract Information paragraph.

ϖ *FileDef.File_Header.Scale*: Scale, part of the standard extract info is used in defining the length and scaling the length to units of bytes. The following constants may be helpful:

Values:

Bits (1/8)	
Nibbles (1/2)	
Bytes = (1)	
Words = (2)	
LongWords (4)	
QuadWords (8)	
[integer >=0]	
Records(-1)	File_Header.Scale can be set to Records(-1). This is a special case that allows the file header to be defined in units of Records using the Record_Header information described in the section.

Example

In the sampleFileDef, the file header is set to a length of one record.

```
%File Header
FileDef.File_Header.Length = 1;
FileDef.File_Header.Scale = Records;
```

4.1 Record Header.

Each record can contain a header. The length of the record and the length of the record header must be specified. If the record length is variable, the length can be stored and retrieved from the record header.

ϖ *FileDef.Record_Header.Header_Length*: This field specifies the length of the Record Header in Bytes.

Values

No_Record_Header (0)	Sets the Record Header length to zero.
[integer >=0]	Sets the Record Header length to the length specified. Units are in Bytes.
Default	This field is required, there is no default.

ϖ *FileDef.Record_Header.Length*: This number indicates the length of a Record.

Values:

From_Header (-1)	DataInsight will read the length from the Record Header
From_File_Header (-3)	A Fixed Record length will be read from file header
[integer >=0]	Sets the Record length to the length specified. Units are in Bytes, unless the FileDef. Record_Header.Scale is set to indicate the units.

Default	From_Header (-1)

ϖ FileDef.Record_Header.Length_Includes_Header: Specifies if the Record length defined by Record_Header.Length includes the header.

Values:	
Yes (1)	Record Length specified includes header.
No (0)	Record Length specified does not include header.
DefaultNo (0)	

FileDef.Record_Header extract info: If Record_Header.Length is set to From_Header(-1), standard extract fields are defined as described in the Standard Extract Information paragraph.

ϖFileDef.Record_Header.Scale: Scale, part of the standard extract info, is used in defining the length and scaling the length to units of bytes. The following constants may be helpful.

Values:
Bits (1/8)
Nibbles (1/2)
Bytes = (1)
Words = (2)
LongWords (4)
QuadWords (8)
[integer >=0]

Example: The sampleFileDef is set up for a record that contains a four-byte header. The record length is read from the record header at word 0 and bit 0 as a 32 bit unsigned integer in units of words. The length that is read from the header does not include the header itself.

```
%Record Header
FileDef.Record_Header.Length = From_Header;
FileDef.Record_Header.Scale = Words;
FileDef.Record_Header.Header_Length = 4;
FileDef.Record_Header.Length_Includes_Header = No;
FileDef.Record_Header.Start_Word = 0;
FileDef.Record_Header.Start_Bit = 0;
FileDef.Record_Header.Field_Length = 32;
FileDef.Record_Header.Byte_Order = Little_Endian;
```

```
FileDef.Record_Header.Data_Type = 'uint32';
```

5.1 Frame Header.

In a similar manner as the record, each frame can contain a header. The length of the frame and the length of the frame header must be specified. If the frame length is variable, the length can be store and retrieved from the frame header. Additionally, the frame length can be specified to be the same as the record length, indicating that there is only frame in the record.

ϖ FileDef.Frame_Header.Header_Length: This field specifies the length of the Record Header in Bytes.

Values:

No_Frame_Header (0)	Sets the Record Header length to zero.
[integer >=0]	Sets the Record Header length to the length specified. Units are in Bytes.
Default	This field is required, there is no default.

ϖ FileDef.Frame_Header.Length: This number indicates the length of a frame.

Values:

From_Frame (-1)	DataInsight will read the length from the Frame Header
From_File_Header (-3)	A Fixed Frame length will be read from file header
[integer >=0]	Sets the Frame length to the length specified. Units are in Bytes, unless the FileDef. Frame_Header.Scale is set to indicate the units.

ϖ same_as_record_Length(-2): Sets the Frame Length equal to the Record Length. This assumes there is one frame/record. The field Frame_Header.Length_Includes_Header should be set to 1 in this case.

Values:

Default	same_as_record_Length(-2)

ϖ FileDef.Frame_Header.Length_Includes_Header: Specifies if the Frame length defined by Frame_Header.Length includes the header.

Values:

Yes (1)	Frame Length specified includes header.
No (0)	Frame Length specified does not include header.
DefaultNo (0)	

ϖ FileDef.Frame_Header extract info: If Frame_Header.Length is set to From_Frame (-1), standard extract fields are defined as described in the Standard Extract Information paragraph.

ϖ FileDef.Frame_Header.Scale: Scale, part of the standard extract info, is used in defining the length and scaling the length to units of bytes. The following constants may be helpful:

Values:
Bits (1/8)
Nibbles (1/2)
Bytes (1)
Words (2)
LongWords (4)
QuadWords (8)
[integer >=0]

Example: The sampleFileDef is set up for a frame length that is the same as the record length, including the fourteen-byte header.

```
%Frame Header
FileDef.Frame_Header.Length = same_as_record_Length;
FileDef.Frame_Header.Header_Length = 14;
FileDef.Frame_Header.Length_Includes_Header = Yes;
```

Frame ID

Each frame can have a unique format and contain unique data. If unique frames are being used, the frame id can be specified within the frame header.

ϖ FileDef.Frame_ID.ID: Number indicating the Frame ID

Values:

From_Frame (-1)	DataInsight will read the length from the frame header.
From_File_Header (-3)	A Frame ID will be read from file header
[integer >=0]	Sets all Frame IDs to the number specified.

FileDef.Frame_ID extract info

If FileDef.Frame_ID.ID is set to From_Frame (-1), standard extract fields are defined as described in the Standard Extract Information paragraph.

Example: The sampleFileDef is set up for a sixteen bit unsigned integer frame id that is stored in word five of the frame header.

```
%Frame ID
FileDef.Frame_ID.ID = From_Frame;
FileDef.Frame_ID.Start_Word = 5;
FileDef.Frame_ID.Start_Bit = 0;
FileDef.Frame_ID.Field_Length = 16;
FileDef.Frame_ID.Byte_Order = Little_Endian;
```

FileDef.Frame_ID.Data_Type = 'uint16';

6.1 Time.

When extracting data, time can be used to specify a range of data to extract. Each frame or record must therefore be associated with a time. This time tag can be in the file or calculated assuming a constant frame rate. Time does NOT have to be monotonically increasing for proper extraction of data within a time range, once the data file has been completely mapped. The data file is automatically completely mapped when a variable is referenced for all time i.e. X{:}.

ϖ *FileDef.Time.Time*:_Type Specified base units for time.

Values:

'time'	Base Unit is seconds
'days'	Base Unit is number of days since A.D. (01 Jan 0000 = 1)

(Note: FileDef.Time.Offset field, part of the standard extraction information, can be used to set another base date)

ϖ *FileDef.Time.Location*: Location of Time in the file

Values:

From_Frame(-1)	Time is read from the Frame Header.
From_Record(~-1)	Time is read from the Record Header
FileDef.Time extract info	Standard extract fields are defined for time as described in the Standard Extract Information paragraph, with the additional information for Data_Type and Time_Offset described here.

ϖ *FileDef Time.Data_Type*: Any legal data type can be used for time as well as the special data types reserved only for time.

Values:

ASCII 'ascii'	Time recorded as an ASCII string conforming to one of the following formats: -days:hours:minutes:seconds.fraction -hours:minutes:seconds.fraction -minutes:seconds.fraction -seconds.fraction

ϖ DATE_TIME_ASCII (*date_time_ASCII*): Time recorded as an ASCII string conforming to one of the following formats:

<u>Probe Compatible formats</u>:

dd-mmm-yyyy-hh:mm:ss
dd-mmm-yy-hh:mm:ss
where the date separator can be a '-', '/', or ' '
date of yy defaults to current century

<u>Other acceptable formats</u>
'dd-mmm-yyyy'
'mm/dd/yy'
'mm/dd'
'HH:MM:SS'
'HH:MM:SS PM'

ϖ CONSTANT_FRAME_RATE: A constant frame rate is assumed using the sampling frequency of FileDef.Time.FrameRate (in Hz) with an offset of FileDef.Time.Offet

Note: IRIG_B (see below) uses time stored in the format at TABLE B-2.

TABLE B-2. CONSTANT FRAME RATE FORMAT

bits	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	sec.				sec*(0.1)				sec*(0.01)				msec			
	Byte 3								Byte 2							
bits	**15**	**14**	**13**	**12**	**11**	**10**	**09**	**08**	**07**	**06**	**05**	**04**	**03**	**02**	**01**	**00**
	Hours *10		Hours				Min * 10			Min				Sec *10		
	Byte 1								Byte 0							

ϖ *STD* *Probe standard format*: Four unsigned integers (16 bit) defined as follows:

uint16 hrs
uint16 sec
uint16 msec
uint16 sec * e-7
Byte order specified by FileDef.Machine_Format

ϖ *FileDef.Time.Offset*: Time offset can be a numeric value specifying an offset to add to the extracted time in units of seconds (for time type of 'time') or days (for time type of 'days'). Time offset can also be specified as a string. If time type = 'days', any of the DATE_TIME_ASCII formats can be used. If time type = 'time', any of the ASCII formats can be used.

Example: The sampleFileDef is set up for a time being stored in the frame header using the Probe STD format.

```
%Time
FileDef.Time.Location = From_Frame;
FileDef.Time.Data_Type = STD_FORMAT;
FileDef.Time.Offset = No_Offset;
FileDef.Time.Scale = 1;
FileDef.Time.Start_Word = 0;
FileDef.Time.Start_Bit = 0;
FileDef.Time.Byte_Order = Little_Endian;
FileDef.Time.Time_Type = 'time';
```

Multiple Time Fields

Since it is common for time to be stored in multiple fields representing hours, minutes, etc. Time can be specified as an array, each with its own scale and location. Each field is read and summed to represent time. For example, the STD format can also be specified as follows:

```
FileDef.Time(1).Location = From_Frame;
FileDef.Time(1).Data_Type = 'uint16';
FileDef.Time(2).Data_Type = 'uint16';
FileDef.Time(3).Data_Type = 'uint16';
FileDef.Time(4).Data_Type = 'uint16';
FileDef.Time(5).Data_Type = 'uint16';
FileDef.Time(1).Scale = HOURS;
FileDef.Time(2).Scale = MINUTES;
FileDef.Time(3).Scale = SECONDS;
FileDef.Time(4).Scale = MILLISEC;
FileDef.Time(5).Scale = 1e-7;
FileDef.Time(1).Start_Word = 0;
FileDef.Time(2).Start_Word = 1;
FileDef.Time(3).Start_Word = 2;
FileDef.Time(4).Start_Word = 3;
FileDef.Time(5).Start_Word = 4;
```

FileDef.Time.Scale

Scale, part of the standard extract info, is used in defining the time and scaling the time to units of seconds. The following constants may be helpful:

MINUTES = 60;
HOURS = 60*60;
SECONDS = 1;
MILLISEC = 0.001;
MICROSEC = 1e-6;
Data Validity Checks

Data insight can verify the integrity of the data file during the mapping operation. Data validation methods include use of a synch word and validation of FrameIDs

- FileDef.Record_Header.SyncData: If set to 1, Data Insight resynchronizes with a synch word at each record boundary. (Default = 0)

- FileDef.Record_Header.Sync.Value
  Value of the synch word

- FileDef.Frame_ID Min_Valid
  Minimum value for Frame_ID allowed. Frames with IDs lower than this value are ignored. (Default 1)
- FileDef.Frame_ID Max_Valid
  Maximum value for Frame_ID allowed. Frames with IDs larger than this value are ignored. (Default inf).
  Frames Spanning Record Boundaries
  By default, Data Insight assumes that frames are fully contained within a record. If frame span record boundaries, this behavior must be specified.

- FileDef.Record_Header. Allow_Split
  If set to 1, frames are allowed to span record boundaries (Default = 0)
  FileDef.Record_Header.SplitFlag extract info
  Standard extract fields are defined for SplitFlag as described in the Standard Extract Information paragraph. The flag should be scaled such that it is greater than one for records that contain frames that span record boundaries.

7.1 Fixed Frame Files

Data Insight allows for a complex record and frame structure. For very simple files containing fixed frame sizes (without a Record/Frame hierarchy), the methods described above are not efficient. For fixed frames, define your frame length as describe below. Synchronization and frame validation do not apply to this case. Unlike the normal case the whole file is mapped at once.

- *FileDef. FixedFrames Header. Length*: Specifies the length of fixed data frames.

8.1 Standard Extract Information

Many of the FileDef descriptions refer to the Standard Extract Information. Data Insight uses the same extract information when specifying a file definition as it does for extracting data. The fields commonly needed for file definition extraction are shown here.

Start_Word: Specifies the location of a variable within a frame, counting from zero in sixteen bit words.
-Word zero is referenced from the start of the file when extracting Source and EventID from the file header.

-Word zero is referenced from the start of the Record Header when extracting Record Length.

-Word zero is referenced from the start of the Frame Header when extracting Frame Length, Frame ID, and Time.

Values
[integer]
Default = 0

Start_Bit: Specifies the *starting* bit of the variable relative to the start word. Bit 0 is considered the least significant bit (LSB) and coincides with the word boundary. Note that Data Insight uses Little Endian representation when specifying bit positions. Big Endian files will be Byte Reversed by default for bit manipulation.

Values:
[integer > 0]
Default = 0

Field_Length: Specifies the length of the variable in bits. When extracting time as in any of the ASCII formats, the field length must be a multiple of eight.

Values:
[integer > 0]
Default = 32

Data_Type: Specifies the data type of the recorded variable. Values are in Table B-3.

TABLE B-3. DATA TYPE OF THE RECORDED VARIABLE		
Data Insight	(C or Fortran equivalent)	Description
'UCHAR'	unsigned char	unsigned character, 8 bits.
'schar'	signed char	signed character, 8 bits.
'int8'	integer*1	integer, 8 bits. Twos complement representation
'int16'	integer*2	integer, 16 bits. Twos complement representation
'int32'	integer*4	integer, 32 bits. Twos complement representation
'int64'	integer*8	integer, 64 bits. Twos complement representation
'uint8'	integer*1	unsigned integer, 8 bits.
'uint16'	integer*2	unsigned integer, 16 bits.
'uint32'	integer*4	unsigned integer, 32 bits.
'uint64'	integer*8	unsigned integer, 64 bits.
'single'	real*4	floating point, 32 bits.
'float32'	real*4	floating point, 32 bits.
'double'	real*8	floating point, 64 bits.
'float64'	real*8	floating point, 64 bits.
'bitN'		signed integer, N bits (1<=N<=64). Twos complement representation
‘ubitN’		unsigned integer, N bits (1<=N<=64).
‘signedmagN’		Signed integer using a sign bit
‘invintN’		Twos complement representation with reversed sign bit.
Note: Default = 'float32'		

Scale: Scale factor to be applied to the raw data after extraction. A scale factor of zero is overriding by a value of 1.

Values:
[any number]
Default = 1

Offset: Offset to be added to the raw data after extraction. The Scale factor is not applied to the Offset.

Values:
[any number]
Default = 0

Bit_Reverse: If set to 1, indicates that Data Insight should reverse the bit order prior to other unpacking routines.

Values (0,1)
Default = 0

Byte_Reverse: If set to 1, indicates that Data Insight should reverse eight bit bytes prior to other unpacking routines. This is performed after bit reversal if Reverse_First = 0 and before bit reversal if Reverse_First = 1. It always occurs prior to word reversal. Note that Data Insight uses Little Endian representation when specifying bit positions. Big Endian files will be Byte Reversed by default for bit manipulation.

Values (0,1)
Default = 0

Reverse_First: Indicates whether Byte reversal should take place after (0) bit reversal or before (1) bit reversal.

Values (0,1)
Default = 0

Word_Reverse: If set to 1, indicates that Data Insight should reverse sixteen bit words prior to other unpacking routines.

Values (0,1)
Default = 0

Setting Default Values: The function setDefaults, will set the default values for all extraction fields. Fields that are defined are left unchanged, undefined fields are set to their default values.

Creating Custom File Mapping Routines: While Data Insight provides a flexible means to interface with data files, some data files may need custom mapping routines. A custom mapping routine is required when the FileDef structure cannot describe the file, or if the performance of the Data Insight mapping routine is not satisfactory.

To use a custom mapping routine, you must first create your own mapping routine using an m-file or a mex file. This file must parse through your data file and identify the following information about each frame of data:

Sample Time of the Frame
Frame ID
Position (in bytes) of the start of frame data
Length of frame (in bytes)

This information should be stored in a four by N array (where N is the number of frames) named FILEMAP.mapfile

Additionally the following information must be set
FILEMAP.status = -1;
FILEMAP.endtime = Last time recorded in the file in seconds.

The FILEMAP structure must be saved in a .MAT format. The data file can then be opened using the openDIData command, with the mapfilename option se

This page intentionally left blank.

APPENDIX C

Outstanding Issues and Technical Challenges

1. Acquiring data sources and DataProbe dictionaries
2. Funding for continued development of Data Insight
3. Acquiring DataProbe

This page intentionally left blank.

APPENDIX D

CURRENT BETA SITES

In order to help determine missing features as well as quality, several sites have already begun using Data Insight. They are:

1. Naval Undersea Warfare Center Division, Keyport[6]
2. Research, Development, and Engineering Center, Redstone Arsenal Army[6]
3. Naval Air Warfare Center, Weapons Division, China Lake
4. Naval Air Warfare Center, Weapons Division, Pt. Mugu
5. Applied Research Lab, Penn State University
6. Tyndall Air Force Base
7. Eglin Air Force Base

The above sites have received a time expiring, P-coded version of Data Insight. The sites have also received associated documentation and an example file.

[6] Has a full subscription to Data Insight

This page intentionally left blank.

INDEX